

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-185359

(43)Date of publication of application : 16.07.1996

(51)Int.Cl.

G06F 12/08

G06F 12/08

G06F 13/36

(21)Application number : 07-151738

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 19.06.1995

(72)Inventor : KOMURO HIROSHI
YAMAGAMI NOBUHIKO
HAYASHI HIROO

(30)Priority

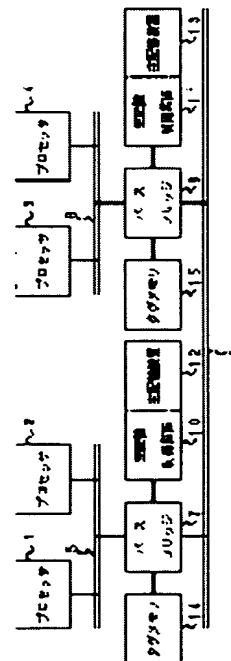
Priority number : 06267770 Priority date : 31.10.1994 Priority country : JP

(54) MEMORY SUBSYSTEM

(57)Abstract:

PURPOSE: To attain coherency control for a cache by the addition of the small number of hardware parts in an NUMA type multilayer cache and to attain high speed latency from a main storage.

CONSTITUTION: Since a bus bridge 14 (15) including a controller for answering the coherency control of the cache by referring to tag information stored in a tag memory 14 (15), and when the tag information indicates 'dirty', stopping data read out from the main memory 12 (13) and sending correct data to a requesting processor 1 (2, 3, 4) after the completion of copy-back processing of a dirty line is constructed, a high speed memory subsystem can be provided by inexpensive hardware.



LEGAL STATUS

[Date of request for examination] 06.03.1997

[Date of sending the examiner's decision of rejection] 28.03.2000

[Kind of final disposal of application other than the examiner's decision of rejection or

application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

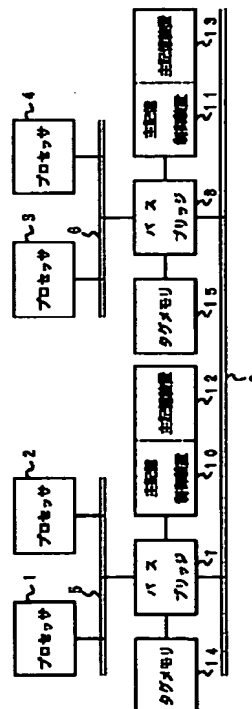
[Number of appeal against examiner's decision of 2000-06093
rejection]

[Date of requesting appeal against examiner's 27.04.2000
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(11)特許出願公開番号



1

【特許請求の範囲】

【請求項 1】 主記憶が分散配置され、独立して動作する内部バスにそれぞれ接続されるキャッシュメモリ内蔵の少なくとも 1 個のプロセッサから成るマルチプロセッサシステムにおいて、

主記憶に実装される全メモリ容量分のタグ情報が記憶されるタグメモリと、

システムバスを介して接続されるプロセッサからのキャッシュの書き替え発生と、主記憶への書き戻しの発生を検出し、上記タグメモリへその状態情報を書き込むタグメモリ制御回路、および当該タグ情報からキャッシュコヒーレンシー制御のレスポンスを内部バスへ出力するとともに、上記タグ情報から主記憶とキャッシュのいずれか正しいレスポンスを選択するコントローラで構成されるバスインターフェイス制御回路とを具備することを特徴とするメモリサブシステム。

【請求項 2】 上記バスインターフェイス制御回路は、キャッシュ内蔵のプロセッサが接続され、独立して動作する複数の内部バスと分散配置される主記憶が接続されるシステムバスの間にあって、

プロセッサからのアクセス要求を受信し、分散配置された主記憶のアドレス範囲を検出し、そのアドレスを内部バスもしくはシステムバスに出力するアドレス比較回路と、

システムバスを常に監視し、以下に示す条件 A、B が成立したとき、メモリタグの情報の変更を行うメモリタグ状態変更回路と、

A. 他方の内部バスに接続されるプロセッサのいずれかで“モデファイ”ステイト（当該キャッシュのみ有効データを持ち主記憶データは無効）が発生したとき、

B. 他方の内部バスに接続されるプロセッサにおいて、“モデファイ”のステートから主記憶へのライトバックが発生したとき、

メモリタグから参照アドレスのタグ情報を得、主記憶から得られるレスポンスデータの正否を判断し、キャッシュコヒーレンシー信号を要求元プロセッサに返却するメモリタグ状態制御回路とで構成されることを特徴とする請求項 1 記載のメモリサブシステム。

【請求項 3】 主記憶が分散配置され、独立して動作する内部バスにそれぞれ接続されるキャッシュメモリ内蔵の少なくとも 1 個のプロセッサから成るマルチプロセッサシステムにおいて、

主記憶に実装される全メモリ容量分のタグ情報が記憶されるタグメモリと、

システムバスを介して接続されるプロセッサからのキャッシュの書き替えの発生と主記憶への書き戻しの発生を検出し、上記タグメモリへその状態情報を書き込むタグメモリ制御回路、および当該タグ情報からキャッシュコヒーレンシー制御のレスポンスを内部バスへ出力する他、上記タグ情報から主記憶へキャッシュの書き戻し完

2

了後、書き替えられたデータを再度読み出しレスポンスを要求元のプロセッサへ返すコントローラで構成されるバスインターフェイス制御回路とを具備することを特徴とするメモリサブシステム。

【請求項 4】 キャッシュ内蔵のプロセッサが接続されたそれぞれ独立して動作するプロセッサバスを一つ以上持ち、これらのプロセッサバスをバスインターフェイス制御回路を介して一つのシステムバスに接続し、主記憶装置が分散されてそれぞれのバスインターフェイス制御回路に接続されるマルチプロセッサシステムにおいて、

上記バスインターフェイス制御回路は、バスに発行されたトランザクションが自身に接続されている自主記憶装置に関係するものであるか否かを判別する比較回路と、

上記自主記憶装置に実装されるメモリ容量分の、データが有効であるか否かを示すタグ情報を記憶するタグメモリと、

上記比較回路による判別結果に基づいて、何れかのプロセッサにより上記自主記憶装置中のデータのキャッシュでの書き換えの発生を検出した時に無効、キャッシュから上記主記憶装置への書き戻しの発生を検出した時に有効を表す状態に、上記タグメモリ中の該当するタグ情報を設定するタグメモリ制御回路と、

上記比較回路により上記自主記憶装置に対する読み出し要求を検出した時に、上記タグメモリの状態情報に応じて、自主記憶装置からデータを読み出して要求に応答するか否かを制御する制御回路とを具備することを特徴とするメモリサブシステム。

【請求項 5】 上記タグ情報に基づいて、プロセッサバスで検出された読み出し要求をシステムバスに発行するかどうかを制御する制御回路を具備することを特徴とする請求項 4 記載のメモリサブシステム。

【請求項 6】 上記タグ情報に基づいて、システムバスで検出された読み出し要求をプロセッサバスに発行するかどうかを制御する制御回路を具備することを特徴とする請求項 4 記載のメモリサブシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、主記憶が分散配置され、独立して動作する内部バスにそれぞれ接続されるキャッシュメモリ内蔵の少なくとも 1 個のプロセッサから成るマルチプロセッサシステムに用いて好適なメモリサブシステムに関する。

【0002】

【従来の技術】 通常、複数のプロセッサを同時動作させるマルチプロセッサシステムにおいて、プログラムを格納する主記憶装置とプロセッサは図 14 のように接続される。図 14 に示す接続構成において、複数の各プロセッサ 41 ~ 4n が一つの主記憶装置 40 を共有すること

3

っても均一のタイミングで応答を返すことが可能であり、一般的にUMA (Uniform Memory Access) と呼ばれている。

【0003】この方式は、システムを容易に構築できる利点がある反面、プロセッサの性能向上に起因するメモリアクセスの集中によるレイテンシの増加や、プロセッサ台数の増加によるプロセッサ、主記憶装置間のバス負荷増加に伴うサイクルタイムの劣化等の欠点が挙げられる。

【0004】上述した欠点を解消するために、図15に示す接続形態が提案された。図15に示す構成において、主記憶装置50a~50nは分散配置されるが、論理的には一つの連続した空間にマッピングされている。

【0005】この場合、一台の主記憶装置に接続されるプロセッサ台数の上限を設定することで高速レイテンシを可能にしている。この構成において、通常の処理はそれぞれのプロセッサ51~5nに接続されている主記憶装置50a~50nのメモリに置かれ、他プロセッサとの同期処理等の必要なアクセスのみ他プロセッサが接続されている主記憶装置内のメモリにアクセスされることが望ましい。

【0006】この方式は、リクエスト先の主記憶装置によりメモリからのレイテンシが異なるため、NUMA (Non Uniform Memory Access) と呼ばれている。NUMA方式のマルチプロセッサシステムにおいては、キャッシュメモリ制御が最大の問題となる。

【0007】マルチプロセッサシステムにおいて、キャッシュメモリは、プロセッサバスをスヌープしてキャッシュコヒーレンシ制御を行うのが一般的であるが、NUMA方式ではプロセッサバス同士が別々に独立して動作しているため、バススヌープ方式を採用することはできない。

【0008】この問題解決のためには、システムバスに大容量のキャッシュを設ける多階層構成キャッシュメモリシステムがある。図16にその接続構成を示す。図中、符号60a~60nはプロセッサユニット、符号61a~61nは主記憶装置、符号62a~62nはキャッシュ、符号63a~63nはインタフェースユニット、符号64a~64nはプロセッサバス、符号65はシステムバスである。

【0009】近年のプロセッサは、プロセッサ内部にプロセッサの周波数と同期している超高速のオンチップキャッシュが数KB程度実装されており、レベル1 (L1) キャッシュと呼ばれている。

【0010】更に、プロセッサの外部に、高速SRAM等のメモリ素子を複数実装しプロセッサが直接アクセスする、数十KB程度のレベル2 (L2) キャッシュが実装される。

【0011】L2キャッシュはL1キャッシュに比べて速度は遅いが、比較的容量が大きいことで知られる。図

4

17にこれらの接続構成が示されている。

【0012】図中、符号71はプロセッサユニットであり、符号72は内蔵のL1キャッシュ、符号73はL2キャッシュコントローラ、符号74はL2キャッシュメモリ、符号75はバスである。

【0013】尚、図16において、単にプロセッサ61a~61nとのみ示されているブロックは図17に示すL1およびL2キャッシュまで含まれていることを前提として以下の説明を行う。

10 【0014】図16に示す例は、更にNUMAの対策用としてレベル3キャッシュ (L3) を設けている。L3キャッシュはプロセッサバス側のL1、L2キャッシュと論理的に同期して動作する。

【0015】つまり、L1またはL2のキャッシュラインの無効化時にはL3キャッシュの同一ラインにも無効化処理を行う。一方、L3キャッシュはシステムバス上のL3キャッシュ間のスヌープ機能を持つことで、全プロセッサのキャッシュのコヒーレンシ制御を実現している。

20 【0016】ここでL3キャッシュはキャッシュの機能を十分に引き出すために通常L2キャッシュの十倍程度かそれ以上の容量が必要である。つまり、L3キャッシュは数十MB程度の容量のメモリ (通常はSRAM) により構成され、上述した機能を実現している。

【0017】

【発明が解決しようとする課題】ところで、上述したNUMA方式の多階層キャッシュは、L3キャッシュの性能を十分に引き出すために大容量 (通常、数十MB程度のSRAM) のメモリが必要で非常に高価なシステム構成となってしまう。

30 【0018】また、L3キャッシュは、プロセッサバス、システムバスのコヒーレンシ制御回路が必要である。更に、プロセッサから主記憶装置内のメモリをリードする場合に、主記憶装置から読み出されたデータは、他プロセッサのキャッシュ内にデータが存在するか否かの条件を他のL3キャッシュからの返事が返ってくるまで正しいか否かを判断することができないといった欠点を有していた。

40 【0019】本発明は上記実情に鑑みなされたもので、NUMA方式の多階層キャッシュにおいて、少量のハードウェアの追加によりキャッシュのコヒーレンシ制御を可能とし、更に、主記憶装置からの高速アクセスを可能にするメモリサブシステムを提供することを目的とする。

【0020】

【課題を解決するための手段】本発明は、主記憶装置が分散配置され、独立して動作する内部バスにそれぞれ接続されるキャッシュメモリ内蔵の少なくとも1個のプロセッサから成るマルチプロセッサシステムにおいて、メモリサブシステムは、主記憶装置に実装される全メモリ

容量分のタグ情報が記憶されるタグメモリと、システムバスを介して接続されるプロセッサからのキャッシュの書き替え発生と、主記憶装置への書き戻しの発生を検出し、上記タグメモリへその状態情報を書き込むタグメモリ制御回路、当該タグ情報からキャッシュコヒーレンシ制御のレスポンスを内部バスへ出力する他、上記タグ情報から主記憶装置とキャッシュのいずれか正しいレスポンスを選択するコントローラで構成されるバスインタフェース制御回路とを具備することを特徴とする。

【0021】尚、上述したコントローラは、当該タグ情報からキャッシュコヒーレンシ制御のレスポンスを内部バスへ出力する他、タグ情報から主記憶装置へキャッシュの書き戻し完了後、書き替えられたデータを再度読み出しレスポンスを要求元のプロセッサへ返す応用も考えられる。

【0022】また本発明は、キャッシュ内蔵のプロセッサが接続されたそれぞれ独立して動作するプロセッサバスを一つ以上持ち、これらのプロセッサバスをバスインターフェイス制御回路を介して一つのシステムバスに接続し、主記憶装置が分散されてそれぞれのバスインターフェイス制御回路に接続されるマルチプロセッサシステムにおいて、上記バスインターフェイス制御回路は、バスに発行されたトランザクションが自身に接続されている自主記憶装置に関係するものであるか否かを判別する比較回路と、上記自主記憶装置に実装されるメモリ容量分の、データが有効であるか否かを示すタグ情報を記憶するタグメモリと、上記比較回路による判別結果に基づいて、何れかのプロセッサにより上記自主記憶装置中のデータのキャッシュでの書き換えの発生を検出した時に無効、キャッシュから上記主記憶装置への書き戻しの発生を検出した時に有効を表す状態に、上記タグメモリ中の該当するタグ情報を設定するタグメモリ制御回路と、上記比較回路により上記自主記憶装置に対する読み出し要求を検出した時に、上記タグメモリの状態情報に応じて、自主記憶装置からデータを読み出して要求に応答するか否かを制御する制御回路とを具備することを特徴とする。

【0023】

【作用】タグメモリには主記憶装置に実装される全メモリ容量分のタグ情報が格納され、システムバスの先に接続されている他方のプロセッサからアクセス要求が発生し、かつキャッシュの書き替えが発生したことが記憶される。バスインタフェース制御回路は、上述したタグ情報を参照してキャッシュのコヒーレンシ制御の返事を行う一方、タグ情報が“ダーティ”を示していた場合、主記憶から読み出されたデータを停止させ、ダーティラインのコピーバック処理の完了を待って正しいデータを要求元のプロセッサに送る。バスインタフェース制御回路は、プロセッサからのアクセス要求を受信し、内蔵するアドレス比較回路にて、分散配置された主記憶のアドレ

ス範囲を検出し、そのアドレスを内部バスもしくはシステムバスに出力する他、システムバスを常に監視し、他方の内部バスに接続されるプロセッサのいずれかで“モディファイ”ステート（当該キャッシュのみ有効データを持ち主記憶データは無効）が発生したとき、あるいは、他方の内部バスに接続されるプロセッサにおいて、“モディファイ”のステートから主記憶へのライトバックが発生したとき、メモリタグ情報の変更を行ない、プロセッサ側では、メモリタグから参照アドレスのタグ情報を得、主記憶から得られるレスポンスデータの正否を判断し、キャッシュコヒーレンシ信号を要求元プロセッサに返却する。

【0024】このことにより、安価なハードウェアで、かつ高速なメモリサブシステムを提供できる。また、各タグメモリがシステム全体のメモリ総容量分のタグ情報（メモリタグ）を記憶するのではなく、同じバスインタフェース制御回路に接続された主記憶装置に実装されたメモリに対応するタグ情報のみをタグメモリに記憶することにより、システム中のメモリ容量の増加に応じて大容量のタグメモリを用意する必要がなくなる。この場合、システム中の何れかのプロセッサのキャッシュにおいて、自主記憶装置のデータに対する書き替えが発生した場合、すなわちバスに無効化要求付きリード要求、あるいは無効化要求のトランザクションが発行された場合には、バスインタフェース制御回路がスヌープして、対象とするデータに対応するメモリタグを無効（Invalid）にし、キャッシュ上のデータが主記憶装置に書き戻された場合、すなわちライト要求のトランザクションが発行された場合に有効（Valid）に設定する。そして、バスからアクセス要求を受け付けた際に、対象とするデータのタグ情報が有効を示すのであれば、自主記憶装置に対するアクセスにより処理を完了し、アクセス要求をバスに発行しない。

【0025】

【実施例】以下図面を参照して本発明の実施例を説明する。図1は本発明の第1実施例によるシステムの構成を示すブロック図である。図1に於いて、符号1～4はプロセッサであり、それぞれが持つプロセッサバス5、6に接続されている。

【0026】プロセッサバス5、6には、更にバスインタフェース制御回路（バスブリッジ）7、8を介してシステムバス9が接続される。バスブリッジ7、8は、プロセッサバス5、6とシステムバス9との間のブリッジ機能と主記憶制御装置10、11とのインタフェース、アドレスとタグとのインタフェース及び制御機能を有する。

【0027】符号12、13はデータを格納する主記憶装置であり、それぞれ主記憶制御装置10、11によりリード／ライト制御がなされる。符号14、15はタグメモリであり、全主記憶分のタグ情報を記憶する。

7

【0028】以下、あるバスブリッジについて、同バスブリッジに直接接続されているプロセッサバスに接続されているプロセッサをローカルプロセッサと呼び、同バスブリッジに直接接続されていないプロセッサバスに接続されているプロセッサをリモートプロセッサと呼ぶ。

【0029】図2は第1実施例において使用されるメモリタグのアドレスマップを示す図であり、図中、21は主記憶アドレスを、22はメモリタグアドレスを示す。タグ中、“0”はリモートプロセッサにダーティライン有り、“1”はリモートプロセッサにダーティライン無しを意味する。

【0030】図3は、図1に示すバスブリッジの回路構成を示す図である。図中、符号31はプロセッサバス入出力制御回路であり、プロセッサバス5(6)とのインタフェース制御を行う。

【0031】符号32はシステムバス入出力制御回路であり、システムバス9とのインタフェース制御を行う。符号33はプロセッサバスアドレス比較回路であり、ここでプロセッサバス5(6)のリクエストアドレスとローカルのメモリアドレスが比較される。

【0032】符号34は主記憶制御入出力回路であり、主記憶制御装置10(11)とのインタフェース制御を行う。符号35はシステムバスアドレス比較回路であり、ここでシステムバス9のリクエストアドレスとローカルのメモリアドレスが比較される。

【0033】符号36はメモリタグステート制御回路であり、ここでメモリタグ情報キャッシュのコヒーレンシーのステート信号を生成する。符号37はメモリタグ制御回路であり、メモリタグとのインタフェース制御を行う。

【0034】符号38はメモリタグステート変更回路であり、ここでシステムバス9のキャッシュスヌープ情報からメモリタグのステート信号を変化させる。以下、本発明の実施例の動作につき、図1乃至図3を使用して説明する。

【0035】図1に示す実施例(システム構成)は、4台のマルチプロセッサシステム構成を例示している。プロセッサ1~4は、それぞれのプロセッサバス5、6に2台ずつ接続されており、それぞれのプロセッサバス5、6は独立して動作している。図示していないが、それぞれのプロセッサ1~4は図17に示すL1、L2のキャッシュメモリによる制御を行っている。

【0036】プロセッサ1からメモリへのアクセスが発生した場合を例にキャッシュコヒーレンシ制御が不要な場合の基本的な動作を説明する。プロセッサ1からのリクエストは、バスブリッジ7に入力される。バスブリッジ7では、受け付けたリクエストが主記憶装置12(この場合のアクセスを“ローカルアクセス”と定義する)の範囲か、それとも主記憶装置13(この場合のアクセスをリモートアクセス”と定義する)への範囲かを判断

8

する。リクエストが主記憶装置12(ローカル)の範囲の場合は、主記憶制御回路10にリクエストが通知され、主記憶制御回路10は主記憶装置12へのアクセスを開始する。ライトリクエストの場合は、ライトデータを主記憶装置12内の指定アドレスへ書き込み、リードリクエストの場合は主記憶装置12からデータを読み出し、逆のバス、即ち、主記憶装置12、バスブリッジ7を経由してプロセッサ1にデータが取り込まれる。

【0037】リクエストが主記憶装置13の範囲(リモート)であった場合は、バスブリッジ7からシステムバス9を介してバスブリッジ8にリクエストが送られる。バスブリッジ8では、上述したローカルアクセスと同様、主記憶制御回路11にリクエストが通知され、主記憶制御回路11は、主記憶装置13をアクセスする。リードリクエストの場合は、バスブリッジ8、システムバス9、バスブリッジ7経由でプロセッサ1にデータが取り込まれる。

【0038】このような機能をバスブリッジ7、8に備えることでプロセッサ1~4は主記憶装置のアドレスを意識することなくリクエストを送出することができる。また、このようなNUMA方式のマルチプロセッサシステムには標準的に含まれている機能である。

【0039】以下の動作は、プロセッサ2にモディファイドラインが存在しなかった場合を仮定している。プロセッサ2はモディファイドラインを持った状態で、そのキャッシュラインに対するリードリクエストをスヌープした時は、「モディファイ」ステータスを要求元プロセッサ1に返却し、そのモディファイドラインのデータをレスポンスデータとして転送する。

【0040】次にキャッシュ制御について説明する。上述したように、それぞれのプロセッサ1~4にはキャッシュが内蔵されている。

【0041】キャッシュのコヒーレンシを保証するために、プロセッサバス1、2はプロセッサバス6の、プロセッサバス3、4はプロセッサバス5のスヌープ機能を実現している。

【0042】ここではコピーバック(またはライトバック)キャッシュ方式のMESIプロトコルをサポートするプロセッサを例にとって説明する。MESIには文字どおり、以下に示す4つのステートが存在する。

【0043】M(モディファイ)：キャッシュライン有効。主記憶データは無効(キャッシュと不一致)当該キャッシュのみ変更データを持つ。

【0044】E(イクスクルーシブ)：キャッシュライン有効。主記憶データは有効(キャッシュと一致)当該キャッシュのみデータが存在する。

【0045】S(シェア)：キャッシュライン有効。主記憶データは有効(キャッシュと一致)。複数のキャッシュにデータが存在する。

【0046】I(インバリッド)：キャッシュライン無

効。主記憶のみ有効。MESIプロトコルのキャッシュでは、プロセッサバスをスヌープし、バスのアクセスから上記4つの状態遷移を行うことで、キャッシュのコヒーレンシ制御を行っている。

【0047】この例では、プロセッサ内部にL2キャッシュコントローラの回路が内蔵されているため、それぞれのプロセッサ1~4が上記キャッシュの制御を行っている。

【0048】しかし、図1に示したシステム構成ではプロセッサバスが2本存在し、それぞれ独立して動作しているため、接続されていないプロセッサバス上のキャッシュのスヌープを行うことは不可能である。

【0049】これを解決するために、メモリタグ14、15とバスブリッジ7、8に以下に示す機能を付加する。まず、メモリタグ14、15について説明を加える。

【0050】図2はメモリタグ14、15のアドレスマッピングを示す。図2に示す例では、主記憶容量64KB、キャッシュラインサイズ64Bの例を示す。

【0051】メモリタグはキャッシュ1ラインに1ビットずつマッピングされる。つまり、“主記憶容量/キャッシュラインサイズ”の容量を持つため、図2に示す例では1Kビット必要となる。

【0052】このメモリタグにはリモートプロセッサにダーティラインが有るか否かの情報が格納されるが、詳細動作については後述する。次に、バスブリッジの機能について説明する。

【0053】バスブリッジ7、8はプロセッサバス5、6とシステムバス9との間のブリッジ機能と、リクエストアドレスを比較する機能があることについては既に述べた。ここでは、バスブリッジ制御の流れについて図3を用いて説明する。

【0054】まず、プロセッサ1からのリクエストがバスブリッジ7内のプロセッサバス入出力制御回路31に入力される。次にリクエストアドレスはプロセッサバスアドレス比較回路33に入力され、主記憶装置12に実装されたメモリ、即ち、ローカルアドレスの場合は主記憶制御入出力回路34に送られる。

【0055】リクエストがローカルアドレスとして存在しない場合は、システムバス入出力制御回路32からシステムバス9に出力される。システムバス9に出力されたリクエストはバスブリッジ8内のシステムバス入出力制御回路32に入力され、更にシステムバスアドレス比較回路35に入力される。

【0056】バスブリッジ8は主記憶装置13のアドレスと比較し、アドレス範囲内の場合にのみ主記憶制御入出力回路34から主記憶回路11に送出される。本発明ではバスブリッジ7、8に更に以下に示す機能が付加される。

【0057】メモリタグステート変更回路38の機能に

ついて述べる。バスブリッジ7、8はシステムバス9を常にスヌープしており、次の条件が成立した時、メモリタグの情報変更要求をメモリタグ制御回路37に送出する。

【0058】“*→1”（リモートダーティ）：リモートプロセッサのいずれかでモディファイステートが発生した時。

“*→0”（リモートクリーン）：リモートプロセッサにおいてモディファイのステートから主記憶装置へのライトバックが発生した時。

【0059】つまり、メモリタグ14、15はシステムバス9からのトランザクションのみ書き換えが行われる。また、ライトバックやモディファイのトランザクション時は、プロセッサバス入出力制御回路31からシステムバス入出力制御回路32へダイレクトのバスで送られる。

【0060】一方、プロセッサバス側では以下の制御が行なわれる。プロセッサからメモリへのトランザクションが発生した場合、プロセッサバス入出力制御回路31からメモリタグ制御回路37にアドレスが通知され、メモリタグ7、8からは図2に示す参照アドレスのタグ情報を読み出す。

【0061】タグ情報はメモリタグステート制御回路36へ送られる。メモリタグステート制御回路36はリードリクエストを受けとった時、メモリタグ情報に基づき以下に示す制御を行う。

(1) メモリタグ=リモートクリーン (“1”) の場合
他方のプロセッサバス6に接続されるプロセッサ3、4のキャッシュラインにモディファイドラインが存在しないため、主記憶装置から読み出されたレスポンスデータが正しいことが判明する。

【0062】このため、プロセッサバス入出力制御回路31からはキャッシュコヒーレンシー制御信号として「シェア」ステータスを要求元プロセッサに返却する。プロセッサ2は「クリーン」または「シェア」ステータスの場合、主記憶装置からのレスポンスデータ待ちとなる。

【0063】ローカルアドレスのリクエストは主記憶装置12から、リモートアドレスに対しては主記憶装置13からレスポンスデータを転送する。また、同一プロセッサバス上に接続されるプロセッサ2が「モディファイ」ステータスの場合は、「モディファイド」ステータスを要求元プロセッサに返却し、プロセッサ2からのモディファイラインのデータがプロセッサ1に転送され、バスブリッジ7は主記憶装置からのデータ転送を禁止する。

(2) メモリタグ=リモートダーティ (“0”) の場合
プロセッサ3または4のキャッシュにダーティラインが存在するため、プロセッサ3または4のキャッシュが読み出される。

【0064】バスブリッジは「シェア」を要求元のプロセッサにステータスとして返す。プロセッサ3または4はダーティラインを主記憶装置に書き戻し、ステータスを「ダーティ」から「シェア」に変化させる。

【0065】バスブリッジは書き戻されたデータを要求元プロセッサ1に送ると同時に、メモリタグのビットを「クリーン」に変化させる。バスブリッジの「シェア」ステータスの返事に対して若干補足説明を加える。

【0066】キャッシュのプロトコルはMESIの4状態をサポートしているが、本発明の方式では「E」（イクスクルーシブ）の状態は無く、MSIの3状態しか存在しない。

【0067】仮りに、主記憶データがあるプロセッサから読み出された場合でも「イクスクルーシブ」でなく「シェア」を書き込む。この処理により、プロセッサは、そのキャッシュラインは他のプロセッサと「シェア」しているものと見せかけている。

【0068】これにより、「イクスクルーシブ」→「モディファイ」のバストランザクションが発生しない状態遷移を禁止している。つまり、キャッシュラインの書き換え（モディファイ）の変化時に必ずプロセッサバスへのバスコヒーレンシ用トランザクションが発生する。このトランザクションが発生するため、メモリタグの制御が可能となる。

【0069】一般的には、キャッシュのコヒーレンシ制御を行う場合、実装されている他の全プロセッサのキャッシュのコヒーレンシの返事を持たなければ、処理を開始することができなかった。

【0070】しかし、本発明の方式を実現することで、モディファイの状態以外の全ての場合において、レスポンスデータが正しいことが保証される。つまり、リクエストがローカルアドレスの場合、システムバスを使用せずにメモリタグのみでトランザクションが完結する。

【0071】尚、本発明実施例では、メモリタグが「ダーティ」の時、ダーティラインの主記憶装置への書き戻しと同時に、プロセッサへのデータ転送とメモリタグ「クリーン」の書き替えを行っているが、制御回路を簡単にするため、ダーティラインの主記憶装置への書き替え後、メモリタグ「クリーン」に変化させ、メモリタグが「クリーン」に書き替えられた後、データを再度主記憶装置から読み出してプロセッサに転送してもよい。つまり、プロセッサからのキャッシュの書き戻しが完了するまで待った後、要求元のプロセッサにレスポンスを返す応用例も考えられる。

【0072】次に、本発明の第2実施例について説明する。第1実施例では、各タグメモリ14、15に、システム中の全主記憶装置に実装される全メモリ容量分のタグ情報（メモリタグ）を格納し、他のプロセッサバスに接続された（リモートプロセッサ）プロセッサによりモディファイステートが発生したことを記憶している。こ

れにより、バスブリッジ7、8は、プロセッサからのアクセス要求に対し、他にダーティなデータが存在する場合にのみシステムバス9にトランザクションを発行するようにできる。

【0073】第2実施例では、第1実施例における効果を確保し、さらにメモリタグを記憶するためのタグメモリの容量を減少させることができるものである。図4は第2実施例に係わるマルチプロセッサシステムの構成を示すブロック図である。図4に示すマルチプロセッサシステムは、NUMA方式の構成となっている。

【0074】図4に示すように、第2実施例のマルチプロセッサは、4つのプロセッサ81～84が設けられている。プロセッサ81、82は、プロセッサバス85を介して接続されている。また、プロセッサ83、84は、プロセッサバス86を介して接続されている。

【0075】各プロセッサ81～84は、それぞれキャッシュ81a～84aを内蔵している（図17中に示すL1、L2のキャッシュメモリ）。第2実施例ではキャッシュメモリラインサイズを64バイトとする。

【0076】プロセッサバス85、86は、それぞれバスインターフェイス制御回路（バスブリッジ）87、88を介して、システムバス89と接続されている。バスブリッジ87、88は、プロセッサバス85、86とシステムバス89との間のブリッジ機能（双方のバス要求の転送機能）と、主記憶制御装置90、91とのインターフェイス、タグメモリ94、95とのインターフェイス及び制御機能を有する。

【0077】主記憶制御装置90、91は、それぞれ主記憶装置92、93に対するメモリアクセス制御（リード制御、ライト制御）を行なう。主記憶装置92、93は、データを格納するものであり、第2実施例ではそれぞれ64Kバイトの容量を持つものとする。

【0078】タグメモリ94、95は、それぞれ各バスブリッジ87、88が管理する主記憶装置92、93に実装されたメモリ容量分のタグ情報（メモリタグ）を記憶する。タグ情報は、キャッシュ1ライン単位ごとに1ビットの情報である。本実施例ではキャッシュ1ライン64バイト、主記憶装置92、93の容量はそれぞれ64Kバイトであるため、タグメモリ94、95の容量はそれぞれ1Kビットの容量となる。

【0079】図5は第2実施例のマルチプロセッサシステムのアドレスマップを示す図であり、0番地から0FFFF番地までが主記憶装置92の主記憶アドレス0番地からFFFF番地に割り当てられ、10000番地から1FFFF番地までが主記憶装置93の主記憶アドレス0番地からFFFF番地に割り当てられている。

【0080】図6は第2実施例において使用されるタグメモリ94（95）のアドレスマップを示す図であり、主記憶アドレス100に示すキャッシュの各管理単位、すなわち各キャッシュラインに対応するメモリタグアド

10

20

30

40

50

レス 102 を示している。

【0081】図 6 に示すように、メモリタグとして、対応するキャッシュラインのデータが無効であるか否かを示す 1 ビットがセットされる。第 2 実施例では、メモリタグ中“0”は主記憶装置中のデータが無効、つまり“Modified”状態のデータが何れかのキャッシュに存在することを示し、“1”は主記憶装置中のデータが有効、つまり“Modified”状態のデータが何れのキャッシュにも存在しないことを示す。

【0082】タグメモリ 94 には主記憶装置 92 に実装されたメモリ分のメモリタグが格納され、タグメモリ 95 には主記憶装置 93 に実装されたメモリ分のメモリタグが格納される。

【0083】図 7 は図 4 に示すバスブリッジ 87 (88) の構成を示すブロック図である。図 7 に示すように、バスブリッジ 87 (88) は、プロセッサバス入出力制御回路 131、システムバス入出力制御回路 132、プロセッサバスアドレス比較回路 133、主記憶制御入出力回路 134、システムバスアドレス比較回路 135、及びタグメモリ入出力回路 136 によって構成されている。

【0084】プロセッサバス入出力制御回路 131 は、プロセッサバスとのインターフェイス制御を行なうもので、プロセッサバスに発行されたトランザクションの入力、プロセッサバスへのトランザクションの発行等を実行する。プロセッサバス入出力制御回路 131 は、入力したトランザクションの、トランザクションが対象とするデータを示すアドレスをプロセッサバスアドレス比較回路 133 に供給する。

【0085】システムバス入出力制御回路 132 は、システムバス 89 とのインターフェイス制御を行なうもので、システムバスに発行されたトランザクションの入力、システムバスへのトランザクションの発行等を実行する。システムバス入出力制御回路 132 は、入力したトランザクションの、トランザクションが対象とするデータを示すアドレスをシステムバスアドレス比較回路 135 に供給する。

【0086】プロセッサバスアドレス比較回路 133 は、プロセッサバス入出力制御回路 131 から供給されたアドレスをもとに、プロセッサバスに発行されたトランザクションが、自バスブリッジに対応する主記憶装置に対するアクセス要求（ローカルアクセス）であるか、あるいは他の主記憶装置に対するアクセス要求（リモートアクセス）であるかを判別する。第 2 実施例では、図 5 に示すように、主記憶装置 92、93 にそれぞれアドレスが割り当てられているので、プロセッサバスアドレス比較回路 133 は、17 ビットで表わされるアドレスの最上位ビットを参照して判別する。バスブリッジ 87 では、アドレスの最上位ビットと“0”とが比較され、バスブリッジ 88 では、最上位ビットと“1”とが比較

される。プロセッサバスアドレス比較回路 133 は、比較した結果、等しい場合にはローカルアドレス、等しくない場合はリモートアドレスと判断する。

【0087】主記憶制御入出力回路 134 は、主記憶制御装置 90 (91) とのインターフェイス制御を行なう。主記憶制御入出力回路 134 は、プロセッサバスアドレス比較回路 133 及びシステムバスアドレス比較回路 135 における比較結果に応じて、主記憶装置 92

(93) に対するアクセスを主記憶制御装置 90 (91) に要求する。なお、主記憶制御入出力回路 134 は、タグメモリ入出力回路 136 による対応するメモリタグの状態に応じて、トランザクション（バス要求）に応じたアクセスを実行する。ただし、主記憶制御入出力回路 134 は、アクセス要求に応じた動作をし実行し、プロセッサバス入出力制御回路 131 またはシステムバス入出力制御回路 132 において、トランザクションに応じた応答を発行するか否かを制御するようにもできる。

【0088】システムバスアドレス比較回路 135 は、システムバス入出力制御回路 132 から供給されるアドレスをもとに、システムバス 89 に発行されたトランザクションが、自バスブリッジに対応する主記憶装置に対するアクセス要求（ローカルアクセス）であるか、あるいは他の主記憶装置に対するアクセス要求（リモートアクセス）であるかを判別する。前述したプロセッサバスアドレス比較回路 133 と同じように、システムバスアドレス比較回路 135 は、17 ビットで表わされるアドレスの最上位ビットを参照して判別する。バスブリッジ 87 では、アドレスの最上位ビットと“0”とが比較され、バスブリッジ 88 では、最上位ビットと“1”とが比較される。プロセッサバスアドレス比較回路 133 は、比較した結果、等しい場合にはローカルアドレス、等しくない場合はリモートアドレスと判断する。

【0089】タグメモリ入出力制御回路 136 は、タグメモリとのインターフェイス制御を行なうもので、プロセッサバス入出力制御回路 131 及びシステムバス入出力制御回路 132 がスヌープしたトランザクションに基づいて、メモリタグの有効 (Valid) / 無効 (Invalid) にセットする。タグメモリ入出力回路 136 は、自バスブリッジに対応する主記憶装置のデータが何れかのプロセッサのキャッシュ上で更新されたことが通知された場合に、該当するキャッシュラインに対応するメモリタグを無効“0”にセットする。また、タグメモリ入出力回路 136 は、キャッシュに格納されていたデータが自バスブリッジに対応する主記憶装置へ書き戻された場合に、該当するキャッシュラインに対応するメモリタグを無効“1”にセットする。

【0090】次に、第 2 実施例におけるキャッシュ制御について説明する。前述したように、それぞれのプロセッサ 81 ~ 84 には、キャッシュ 81a ~ 84a が内蔵されている。プロセッサ 81 ~ 84 は、キャッシュのコ

ヒーレンシを保証するために、プロセッサバスをスヌープし、キャッシュに格納されたデータの状態を管理している。なお、キャッシュ81a~84aは、コピーバックキャッシュ方式を用いているものとする。

【0091】第2実施例ではプロセッサのキャッシュは、Modified/Shared/Invalidの3状態(MSI)で管理されるものとする。第2実施例のマルチプロセッサシステムでは、プロセッサによるキャッシュ上でのデータ変更が発生した場合、データ変更の発生がバスブリッジに通知(スヌープ)され、タグメモリに反映される必要がある。しかし、MESIプロトコルに基づいて制御されるキャッシュでは、プロセッサがExclusive状態と呼ばれる状態のキャッシュメモリの書き換えを行った場合、データの変更が外部に通知されず、タグメモリに反映されなくなってしまう。

【0092】一般に、MSIの3状態に、Exclusive状態を加えた4状態でキャッシュメモリを管理するプロセッサが多く使われている。この様なプロセッサを使用する場合でも、プロセッサがリード要求を発行する度にバス上に共有(Shared)応答を返す回路を付加することにより(バスブリッジ(バスインターフェイス制御回路)に付加することも可能)、疑似的にMSIの3状態のみを持つようにすることができる。すなわち、プロセッサは、リード要求の発行に対して共有応答が返されると、読み込んだデータをShared状態として管理する。これにより、このデータを更新する際に、データ変更の発生を外部に通知するようになり、一般的なMESIプロトコルを用いたプロセッサを用いても、疑似的にMSIの3状態で管理するマルチプロセッサシステムとすることができる。

【0093】図8に3状態のキャッシュの状態遷移図を示している。各状態の意味、各状態でロード(Load)命令、ストア(store)命令を実行した時の処理は以下の通りである。

【0094】(A) Invalid (I: 無効)

(a1) そのキャッシュラインのデータが無効であることを示す。

(a2) キャッシュメモリ有効にした時の初期値。

【0095】(a3) ロード命令を実行するとロードミスとなり、バスにリード要求を発行し、Shared状態へ遷移する。

(a4) ストア命令を実行するとストアミスとなり、バスに無効化つきリード要求を発行し、Modified状態へ遷移する。

【0096】(B) Shared (S: 共有)

(b1) そのキャッシュラインのデータが有効であることを示す。その内容はメモリに格納されている物と一致する。

【0097】(b2) ロード命令を実行するとロードヒ

ットとなり、バス要求も状態遷移も起きない。

(b3) ストア命令を実行するとストアヒットとなり、バスに無効化要求を発行しModified状態へ遷移する。

【0098】(b4) キャッシュラインの置き換えが起きた時、およびバス上の無効化要求もしくは無効化つきリード要求をスヌープした時は、バス要求を発行せずにInvalid状態へ遷移する。

【0099】(b5) リード要求をスヌープした時は、バス要求も状態遷移も起きない。

(C) Modified (M: 変更済み)

(c1) そのキャッシュラインのデータが有効であることを示す。メモリに格納されているデータは無効である。(つまりこのキャッシュラインに格納されているデータのみが有効)

(c2) ロード命令を実行するとロードヒットとなり、バス要求も状態遷移も起きない。

【0100】(c3) ストア命令を実行するとストアヒットとなり、バス要求も状態遷移も起きない。

(c4) キャッシュラインの置き換えが起きた時、およびバス上の無効化つきリード要求をスヌープした時は、バスにライト要求を発行しメモリにデータを書き戻し、Invalid状態へ遷移する。

【0101】(c5) リード要求をスヌープした時は、バスにライト要求を発行しメモリにデータを書き戻し、Shared状態へ遷移する。

(c6) あるプロセッサがこの状態の時は、他の全てのプロセッサは全てInvalid状態である。従って無効化要求をスヌープすることはない。

【0102】次に、第2実施例におけるメモリタグの制御について説明する。図9にメモリタグの状態遷移図を示している。メモリタグは、キャッシュライン毎に1ビットの情報を持つ。つまり2つの状態(Valid、Invalid)を持ち、それぞれ次のような意味を持つ。

【0103】Valid (有効)。システム中の何れのプロセッサも対応するデータをキャッシュにModified状態で持っていない。つまり、主記憶装置に格納されているデータが有効なデータである。

【0104】Invalid (無効)。システム中の何れかのプロセッサが、対応するデータをキャッシュにModified状態で持っている。つまり主記憶装置に格納されているデータは無効なデータである。

【0105】キャッシュの使用を開始する時点では、何れのプロセッサのキャッシュも全てInvalid状態であるため、メモリタグの値は全てValidとなっている。

【0106】図8に示すキャッシュの状態遷移の中で、Modified状態へ遷移する時、また、この時のみ、必ずバスに無効化要求か無効化つきリード要求が発

行される。また、Modified状態から他の状態へ遷移する時、またこの時のみ、必ずバスにライト要求が発行される。従って、バス上に無効化要求もしくは無効化つきリード要求が発行された時にメモリタグをInvalidに変更し、バス上にライト要求が発行された時にメモリタグをValidに変更することにより、メモリタグの内容をその定義した値に維持することが可能となる。

【0107】次に、第2実施例の動作について説明する。ここでは、プロセッサ81から(1)リード要求、(2)無効化付きリード要求、(3)無効化要求、(4)ライト要求、が発生した場合についての動作をそれぞれ説明する。

【0108】(1) リード要求

プロセッサ81がリード要求を発行した場合の処理の流れを、図10のフローチャートに示している。なお、フローチャート中に示す括弧内の数字は、その処理を実行する機能ブロックの図4及び図7における参照符号を示している。

【0109】まず、プロセッサ81は、リード要求のトランザクションをプロセッサバス85に発行する(ステップA1)。ここで、プロセッサバス85に接続された他のプロセッサ82が、要求されたデータをModified状態でキャッシュ82aに持っている場合(ステップA2)、プロセッサ82は、リトライ応答を行ない、リード要求を一度中断させる(ステップA3)。プロセッサ82は、要求されたModified状態のデータを主記憶装置へコピーバックする(これはスヌープ型プロセッサの通常の動作である)。その後、プロセッサ81は、リード要求を再発行する。

【0110】バスブリッジ87は、プロセッサバス85に発行されたリード要求を、プロセッサバス入出力制御回路131で取り込む(ステップA4)。前述したステップA2、A3のように、共通するプロセッサバス85に接続された他のプロセッサによってリード要求に対するリトライ応答を受けた場合は、その時点でリード要求を破棄する。

【0111】バスブリッジ87は、プロセッサバスアドレス比較回路133を用いて、受け付けたリード要求が主記憶装置92の領域の範囲(ローカルアクセス)か、あるいは、それ以外(図4の構成では主記憶装置93の領域の範囲(リモートアクセス)かを、対象とするデータを示すアドレス(キャッシュラインアドレス)を参照して判別する(ステップA5)。

【0112】一方、タグメモリ入出力回路136は、タグメモリ94から、リード要求の対象アドレスのメモリタグの値を読み出し、主記憶装置92中のデータが有効(Valid)であるか否かを判別する(ステップA6)。

【0113】なお、メモリタグの値を判別する処理は、

ローカルアクセスの場合にのみ意味があるので、プロセッサバスアドレス比較回路133によって、リード要求がローカルアクセスであると判別されてからタグメモリの読み出しを開始すれば良い。この際、タグメモリ入出力回路136は、プロセッサバスアドレス比較回路133からの判別結果の通知を受けて、タグメモリ94に対するアクセスを開始する。ただし、高速化のために、タグメモリ入出力回路136は、プロセッサバスアドレス比較回路133による判別結果を待たず、プロセッサバス入出力制御回路131からのリード要求受け付けの通知(対象とするアドレスを含む)に応じて、タグメモリの読み出し、及び判別を開始するようにしても良い。

【0114】プロセッサバスアドレス比較回路133によるアドレス比較の結果がリモートアクセスであった場合、もしくはメモリタグが無効(Invalid)を示す場合には、システムバス入出力制御回路132は、プロセッサバス入出力制御回路131を介して得たリード要求をシステムバス89に発行する(ステップA7)。

【0115】一方、プロセッサバスアドレス比較回路133によるアドレス比較の結果がローカルアクセスであり、かつメモリタグが有効(Valid)を示す場合には、主記憶制御入出力回路134は、主記憶制御装置90にリード要求を通知する。主記憶制御装置90は、主記憶装置92に対してリードアクセスを実行し、該当するデータを読み込み、主記憶制御入出力回路134に出力する。読み出されたデータは、主記憶制御入出力回路134、プロセッサバス入出力制御回路131、プロセッサバス85を介して、プロセッサ81に供給される(ステップA8)。

【0116】なお、タグメモリ94へのアクセスと同様に、高速化のために、主記憶制御入出力回路134は、プロセッサバスアドレス比較回路133でのアドレス比較と、タグメモリ入出力回路136によるメモリタグの判別を待たずに、主記憶装置92に対する読み出し制御を開始して良い。

【0117】また、主記憶制御入出力回路134は、メモリタグの状態に関係なくリード要求に応じて主記憶装置92に対してアクセス制御し、プロセッサバス入出力制御回路131がタグメモリ入出力回路136による判別結果に応じて、プロセッサバス85に出力するか否かを制御しても良い。

【0118】ところで、ステップA7において、システムバス89に発行されたリード要求は、バスブリッジ88のシステムバス入出力制御回路132で取り込まれる(ステップA9)。

【0119】バスブリッジ88は、システムバスアドレス比較回路135を用いて、受け付けたリード要求が主記憶装置93の領域の範囲(ローカルアクセス)か、あるいは、それ以外(リモートアクセス)かを、対象とするデータを示すアドレス(キャッシュラインアドレス)

10

20

30

40

50

を参照して判別する（ステップA10）。第2実施例では、説明を簡単にするために、システムバス89に2つのバスブリッジ87、88のみが接続された構成として、ここでは必ずローカルアクセスとなる。従って、ステップA10の処理は不要であるが、システムバス89に他のバスブリッジやI/Oデバイスなどが接続される場合には必要となる。

【0120】一方、タグメモリ入出力回路136は、タグメモリ95から、リード要求の対象アドレスのメモリタグの値を読み出し、主記憶装置93中のデータが有効（Valid）であるか否かを判別する（ステップA11）。

【0121】なお、メモリタグの値を判別する処理は、ローカルアクセスの場合にのみ意味があるので、システムバスアドレス比較回路135によって、リード要求がローカルアクセスであると判別されてからタグメモリの読み出しを開始すれば良い。この際、タグメモリ入出力回路136は、システムバスアドレス比較回路135からの判別結果の通知を受けて、タグメモリ95に対するアクセスを開始する。ただし、高速化のために、タグメモリ入出力回路136は、システムバスアドレス比較回路135による判別結果を待たず、システムバス入出力制御回路132からのリード要求受け付けの通知（対象とするアドレスを含む）に応じて、タグメモリの読み出し、及び判別を開始するようにしても良い。

【0122】システムバスアドレス比較回路135によるアドレス比較の結果がリモートアクセスであった場合、もしくはメモリタグが無効（Invalid）を示す場合には、プロセッサバス入出力制御回路131は、システムバス入出力制御回路132を介して得たリード要求をプロセッサバス86に発行する（ステップA12）。

【0123】一方、プロセッサバスアドレス比較回路133によるアドレス比較の結果がローカルアクセスであり、かつメモリタグが有効（Valid）を示す場合には、主記憶制御入出力回路134は、主記憶制御装置91にリード要求を通知する。主記憶制御装置91は、主記憶装置93に対してリードアクセスを実行し、該当するデータを読み込み、主記憶制御入出力回路134に出力する。読み出されたデータは、（バスブリッジ88の主記憶制御入出力回路134及びシステムバス入出力制御回路132、システムバス89、バスブリッジ87のシステムバス入出力制御回路132及びプロセッサバス入出力制御回路131、プロセッサバス85を介して、プロセッサ81に供給される（ステップA13）。

【0124】なお、タグメモリ95へのアクセスと同様に、高速化のために、主記憶制御入出力回路134は、システムバスアドレス比較回路135でのアドレス比較と、タグメモリ入出力回路136によるメモリタグの判別を待たずに、主記憶装置93に対する読み出し制御を

開始して良い。

【0125】また、主記憶制御入出力回路134は、メモリタグの状態に関係なくリード要求に応じて主記憶装置93に対してアクセス制御し、システムバス入出力制御回路132がタグメモリ入出力回路136による判別結果に応じて、システムバス89に出力するか否かを制御しても良い。

【0126】ところで、ステップA12において、プロセッサバス86に発行されたリード要求は、プロセッサ83、84によってスヌープされる。プロセッサ83、84は、自キャッシュ83a、84aにリード要求のあったデータを、Modified状態で保持していないかを調べる。もし、あった場合はリトライ応答を行い、リード要求を一度中断させ、該当するデータを主記憶装置へコピーバックする（ステップA14）。

【0127】プロセッサ81は、リード要求をリトライし、その結果、主記憶装置にコピーバックされたデータを読み取ることができる。

（2）無効化付きリード要求

プロセッサ81が無効化リード要求を発行した場合の処理の流れを、図11のフローチャートに示している。図10と同様に、フローチャート中に示す括弧内の数字は、その処理を実行する機能ブロックの図4及び図7における参照符号を示している。

【0128】無効化付きリード要求の処理は、前述したリード要求の場合と大きな処理の流れが同じであるため、ここでは異なる点に着目して説明する。前述したリード要求時の処理では、プロセッサバスアドレス比較回路133（バスブリッジ87）またはシステムバスアドレス比較回路135（バスブリッジ88）におけるアドレス比較の結果、リモートアクセスであったと判別された場合、もしくはタグメモリ入出力回路13による対象とするデータに対応するメモリタグが無効（Invalid）であった場合にのみ、入力したリード要求を他方のバスに発行している。すなわち、バスブリッジ87は、前述した条件の場合にのみ、プロセッサバス85からのリード要求をシステムバス89に発行する。またバスブリッジ88は、同様にシステムバス89からのリード要求をプロセッサバス86に発行する。

【0129】これに対し、無効化付きリード要求の場合は、主記憶装置からのデータの読み出しの他に、システム中の全てのキャッシュの該当するデータを無効化するため、全てのプロセッサに無効化付きリード要求をスヌープさせる必要がある。

【0130】このため、バスブリッジ87は、プロセッサバス入出力制御回路131によって無効化付きリード要求を受け付けると（ステップB4）、無条件にシステムバス入出力制御回路132を介してシステムバス89に無効化付きリード要求を発行する（ステップB9）。

【0131】同様にして、バスブリッジ88は、システ

ムバス入出力制御回路 1 3 2 によって無効化付きリード要求を受付けると（ステップ B 1 0）、無条件にプロセッサバス入出力制御回路 1 3 1 を介してプロセッサバス 8 6 に無効化付きリード要求を発行する（ステップ B 1 5）。

【0 1 3 2】バスブリッジ 8 7 は、プロセッサバスアドレス比較回路 1 3 3 によるアドレス比較の結果がリモートアクセスであった場合、もしくはメモリタグが無効（Invalid）を示す場合には、無条件に無効化付きリード要求がシステムバス 8 9 に発行されているので処理を終了する（ステップ B 5、B 6）。

【0 1 3 3】また、バスブリッジ 8 7 は、プロセッサバスアドレス比較回路 1 3 3 によるアドレス比較の結果がローカルアクセスであり、かつメモリタグが有効（Valid）を示す場合には、主記憶装置 9 2 から該当するデータを読み出して、プロセッサバス 8 5 を介してプロセッサ 8 1 に供給する（ステップ B 7）。また、バスブリッジ 8 7 は、タグメモリ入出力回路 1 3 6 によって、該当するデータに対応するメモリタグの値が有効（Valid）を示す場合には、ローカルアクセス、リモートアクセス何れの場合でも、無効（Invalid）に書き換える（ステップ B 8）。

【0 1 3 4】バスブリッジ 8 8 は、システムバスアドレス比較回路 1 3 5 によるアドレス比較の結果がリモートアクセスであった場合、もしくはメモリタグが無効（Invalid）を示す場合には、無条件に無効化付きリード要求がプロセッサバス 8 6 に発行されているので処理を終了する（ステップ B 1 1、B 1 2）。

【0 1 3 5】また、バスブリッジ 8 8 は、システムバスアドレス比較回路 1 3 5 によるアドレス比較の結果がローカルアクセスであり、かつメモリタグが有効（Valid）を示す場合には、主記憶装置 9 3 から該当するデータを読み出して、システムバス 8 9、バスブリッジ 8 7 を介してプロセッサ 8 1 に供給する（ステップ B 1 3）。また、バスブリッジ 8 8 は、タグメモリ入出力回路 1 3 6 によって、該当するデータに対応するメモリタグの値が有効（Valid）を示す場合には、無効（Invalid）に書き換える（ステップ B 1 5）。

【0 1 3 6】ところで、ステップ B 1 5 において、プロセッサバス 8 6 に発行された無効化付きリード要求は、プロセッサ 8 3、8 4 によってスヌープされる。プロセッサ 8 3、8 4 は、自キャッシュ 8 3 a、8 4 a にリード要求のあったデータを、Modified 状態で保持しているかを調べる。もし、あった場合はリトライ応答を行い、リード要求を一度中断させ、該当するデータを主記憶装置へコピーバックする（ステップ B 1 5）。また、プロセッサ 8 3、8 4 は、自キャッシュ 8 3 a、8 4 a の該当するデータの状態を無効（Invalid）に変更する。

【0 1 3 7】（3）無効化要求

プロセッサ 8 1 が無効化要求を発行した場合の処理の流れを、図 1 2 のフローチャートに示している。図 1 0 と同様に、フローチャート中に示す括弧内の数字は、その処理を実行する機能ブロックの図 4 及び図 7 における参照符号を示している。

【0 1 3 8】無効化要求の処理は、前述した無効化付きリード要求と大きな処理の流れが同じであるため、ここでは異なる点に着目して説明する。無効化要求はキャッシュ 8 1 a に格納されたデータが共有（Shared）状態である時に発行されるため、他のプロセッサ 8 2、8 3、8 4 のキャッシュ 8 2 a、8 3 a、8 4 a では Modified 状態でありえない（あるプロセッサのキャッシュで Modified 状態であるデータは、他のプロセッサのキャッシュにおいて Invalid 状態でなければならない）。そのため、プロセッサバス 8 5、8 6 に接続された他のプロセッサ 8 2、8 3、8 4 では、キャッシュに Modified 状態のデータを持つことによるリトライ応答は発生しない。

【0 1 3 9】バスブリッジ 8 7 は、プロセッサバス入出力制御回路 1 3 1 によって無効化要求を受付けると（ステップ C 2）、無条件にシステムバス入出力制御回路 1 3 2 を介してシステムバス 8 9 に無効化要求を発行する（ステップ C 5）。

【0 1 4 0】同様にして、バスブリッジ 8 8 は、システムバス入出力制御回路 1 3 2 によって無効化要求を受付けると（ステップ C 6）、無条件にプロセッサバス入出力制御回路 1 3 1 を介してプロセッサバス 8 6 に無効化要求を発行する（ステップ C 9）。

【0 1 4 1】バスブリッジ 8 7 は、プロセッサバスアドレス比較回路 1 3 3 によるアドレス比較の結果がリモートアクセスであった場合には、無条件に無効化要求がシステムバス 8 9 に発行されているので処理を終了する（ステップ C 3）。また、バスブリッジ 8 7 は、プロセッサバスアドレス比較回路 1 3 3 によるアドレス比較の結果がローカルアクセスである場合には、タグメモリ入出力回路 1 3 6 によって、該当するデータに対応するメモリタグの値を無効（Invalid）に書き換える（ステップ C 4）。

【0 1 4 2】バスブリッジ 8 8 は、システムバスアドレス比較回路 1 3 5 によるアドレス比較の結果がリモートアクセスであった場合には、無条件に無効化要求がプロセッサバス 8 6 に発行されているので処理を終了する（ステップ C 7）。また、バスブリッジ 8 8 は、システムバスアドレス比較回路 1 3 5 によるアドレス比較の結果がローカルアクセスである場合には、タグメモリ入出力回路 1 3 6 によって、該当するデータに対応するメモリタグの値を無効（Invalid）に書き換える（ステップ C 8）。

【0 1 4 3】プロセッサバス 8 6 に発行された無効化要求は、プロセッサ 8 3、8 4 によってスヌープされる。

プロセッサ83, 84は、自キャッシュ83a, 84a中に該当するデータが存在すれば、データの状態を無効(Invalid)に変更する。

【0144】(4)ライト要求

プロセッサ81がライト要求を発行した場合の処理の流れを、図13のフローチャートに示している。図10と同様に、フローチャート中に示す括弧内の数字は、その処理を実行する機能ブロックの図4及び図7における参照符号を示している。

【0145】ライト要求の処理は、前述した無効化付きリード要求の場合と大きな処理の流れが同じであるため、ここでは異なる点に着目して説明する。ライト要求はキャッシュ81aがModified状態である時に発行されるため、他のプロセッサ82, 83, 84のキャッシュ82a, 83a, 84aでは全てInvalid状態である。そのため、プロセッサバス85, 86に接続された他のプロセッサ82, 83, 84では、キャッシュにModified状態のデータを持つことによるリトライ応答は発生しない。

【0146】バスブリッジ87は、プロセッサバス85に発行されたライト要求を、プロセッサバス入出力制御回路131で取り込む(ステップD2)。バスブリッジ87は、プロセッサバスアドレス比較回路133を用いて、受け付けたライト要求がローカルアクセスか、あるいはリモートアクセスかを、対象とするデータを示すアドレス(キャッシュラインアドレス)を参照して判別する(ステップD3)。

【0147】プロセッサバスアドレス比較回路133によるアドレス比較の結果がリモートアクセスであった場合、システムバス入出力制御回路132は、プロセッサバス入出力制御回路131を介して得たライト要求をシステムバス89に発行する(ステップD6)。

【0148】一方、プロセッサバスアドレス比較回路133によるアドレス比較の結果がローカルアクセスである場合には、主記憶制御入出力回路134は、主記憶制御装置90にライト要求を通知する。主記憶制御装置90は、主記憶装置92に対してライトアクセスを実行し、該当するデータを書き込む。また、タグメモリ入出力回路13は、ライトアクセスの対象となるデータに対応するメモリタグを有効(Valid)にする(ステップD5)。なお、タグメモリ入出力回路13によるメモリタグの変更と、主記憶制御入出力回路134による主記憶装置92に対するライト要求は、並行して実行することができる。

【0149】ところで、ステップD6において、システムバス89に発行されたライト要求は、バスブリッジ88のシステムバス入出力制御回路132で取り込まれる(ステップD7)。

【0150】バスブリッジ88は、システムバスアドレス比較回路135を用いて、受け付けたライト要求がロ

ーカルアクセスか、あるいはリモートアクセスかを、対象とするデータを示すアドレス(キャッシュラインアドレス)を参照して判別する(ステップD8)。

【0151】システムバスアドレス比較回路135によるアドレス比較の結果がリモートアクセスであった場合、処理を終了する。また、システムバスアドレス比較回路135によるアドレス比較の結果がローカルアクセスである場合には、主記憶制御入出力回路134は、主記憶制御装置90にライト要求を通知する。主記憶制御装置90は、主記憶装置92に対してライトアクセスを実行し、該当するデータを書き込む。また、タグメモリ入出力回路13は、ライトアクセスの対象となるデータに対応するメモリタグを有効(Valid)にする(ステップD10)。なお、タグメモリ入出力回路13によるメモリタグの変更と、主記憶制御入出力回路134による主記憶装置92に対するライト要求は、並行して実行することができる。

【0152】このようにして、バスブリッジ(バスインターフェイス制御回路)87, 88は、バス上に発行されたリード要求もしくは無効化付きリード要求を検出したとき、その対象となるメモリアドレスがローカルアクセスであるか判別し、ローカルアクセスであった場合はメモリタグの値を調べる。この時、メモリタグの値が有効(Valid)を示す場合は主記憶装置からデータを読み出してリード要求に応答する。すなわち、システム中の他のプロセッサからの通知等を待つことなく、処理を完了することができる。もし、無効(Invalid)を示す場合は、Modified状態のデータをもったプロセッサが、他に必ず存在するため、システムバスにトランザクションを発行することで、該当するプロセッサから応答させる。

【0153】以上のようにして、階層をもったスヌープ方式のキャッシュコヒーレンシ制御と、主記憶装置からの高速なレイテンシを提供することができる。しかも、タグメモリ94, 95に必要な記憶容量は、それぞれに対応する主記憶装置のみに関するメモリタグ分で良く、第1実施例と比較すると大幅に少なくすることができる。

【0154】

【発明の効果】以上詳記したように本発明によれば、タグメモリに格納されたタグ情報を参照してキャッシュのコヒーレンシ制御の返事を行う一方、タグ情報が“ダーティ”を示していた場合、主記憶から読み出されたデータを停止させ、ダーティラインのコピーバック処理の完了を待って正しいデータを要求元のプロセッサに送るコントローラ内蔵のバスブリッジを提供することにより、安価なハードウェアで、かつ高速メモリサブシステムを構築できる。このことにより、L3キャッシュのような大規模なキャッシュメモリが不要となり、更に他方のL3キャッシュのコヒーレンシ制御の返事待つことなし

に主記憶から読み出されたデータを要求プロセッサに高速に転送可能となる。

【図面の簡単な説明】

【図 1】 本発明の第 1 実施例の構成を示すブロック図。

【図 2】 第 1 実施例にて使用されるメモリタグのアドレスマッピングの例を示す図。

【図 3】 図 1 に示すバスブリッジの内部構成を示すブロック図。

【図 4】 第 2 実施例に係わるマルチプロセッサシステムの構成を示すブロック図。

【図 5】 第 2 実施例のマルチプロセッサシステムのアドレスマップを示す図。

【図 6】 図 4 に示すタグメモリ 9 4 (9 5) のアドレスマップを示す図。

【図 7】 図 4 に示すバスブリッジ 8 7 (8 8) の構成を示すブロック図。

【図 8】 第 2 実施例における 3 状態 (MSI) のキャッシュの状態遷移を説明するための図。

【図 9】 第 2 実施例におけるメモリタグの状態遷移を説明するための図。

【図 10】 リード要求を発行した場合の処理の流れを示すフローチャート。

【図 11】 無効化付きリード要求を発行した場合の処理の流れを示すフローチャート。

【図 12】 無効化要求を発行した場合の処理の流れを示

すフローチャート。

【図 13】 ライト要求を発行した場合の処理の流れを示すフローチャート。

【図 14】 均等メモリアクセス方式採用のマルチプロセッサシステムの構成例を示す図。

【図 15】 不均等メモリアクセス方式採用のマルチプロセッサシステムの構成例を示す図。

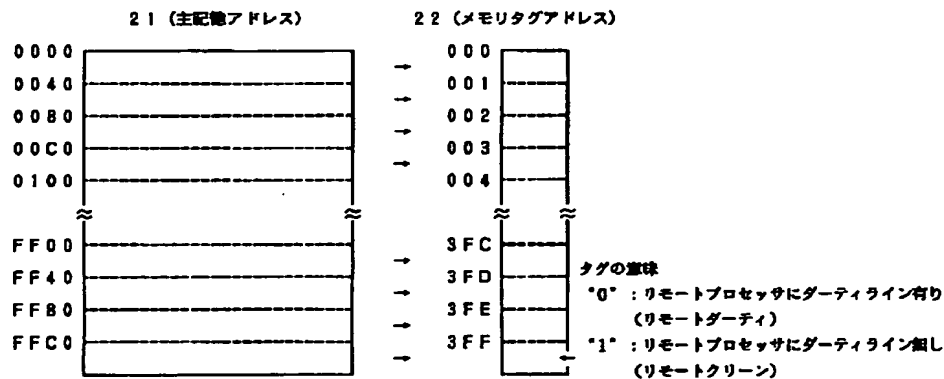
【図 16】 多階層キャッシュを採用したマルチプロセッサシステムの構成例を示す図。

10 【図 17】 プロセッサ内キャッシュの構成例を示す図。

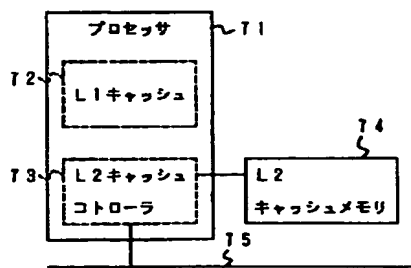
【符号の説明】

1, 2, 3, 4, 8 1, 8 2, 8 3, 8 4…プロセッサ、5, 6, 8 5, 8 6…プロセッサバス (内部バス)、7, 8, 8 7, 8 8…バスブリッジ、9, 8 9…システムバス、10, 11, 9 0, 9 1…主記憶制御装置、12, 13, 9 2, 9 3…主記憶装置、14, 15, 9 4, 9 5…タグメモリ、3 1, 1 3 1…プロセッサバス入出力制御回路、3 2, 1 3 2…システムバス入出力制御回路、3 3, 1 3 3…プロセッサバスアドレス比較回路、3 4, 1 3 4…主記憶制御入出力回路、3 5, 1 3 5…システムバスアドレス比較回路、3 6…メモリタグステート制御回路、3 7…メモリタグ制御回路、3 8…メモリタグステート変更回路、1 3 6…タグメモリ入出力回路。

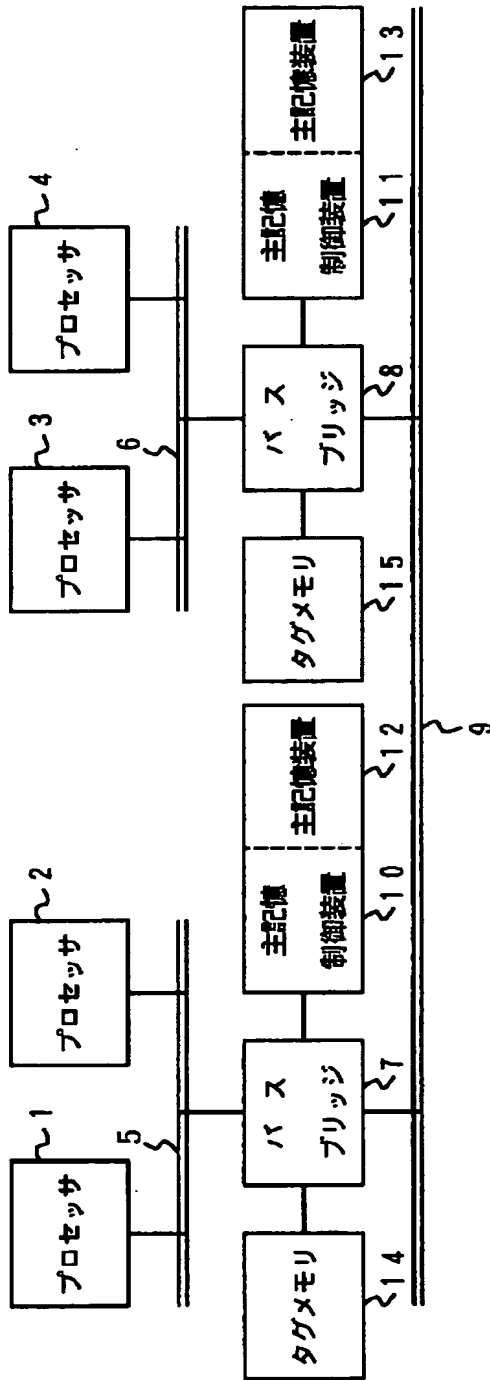
【図 2】



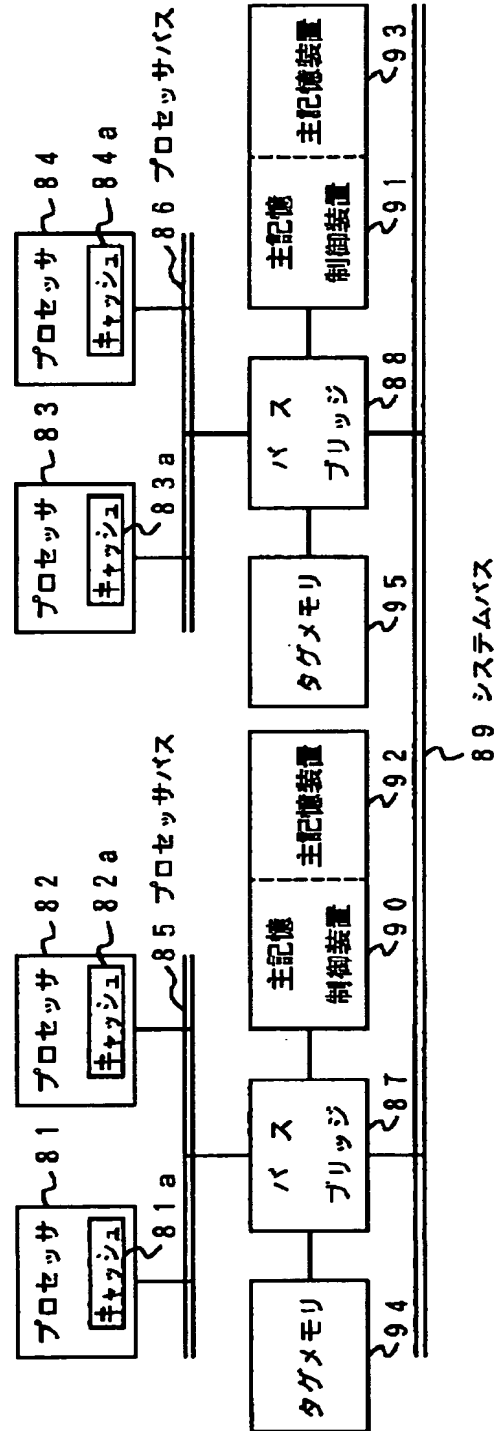
【図 17】



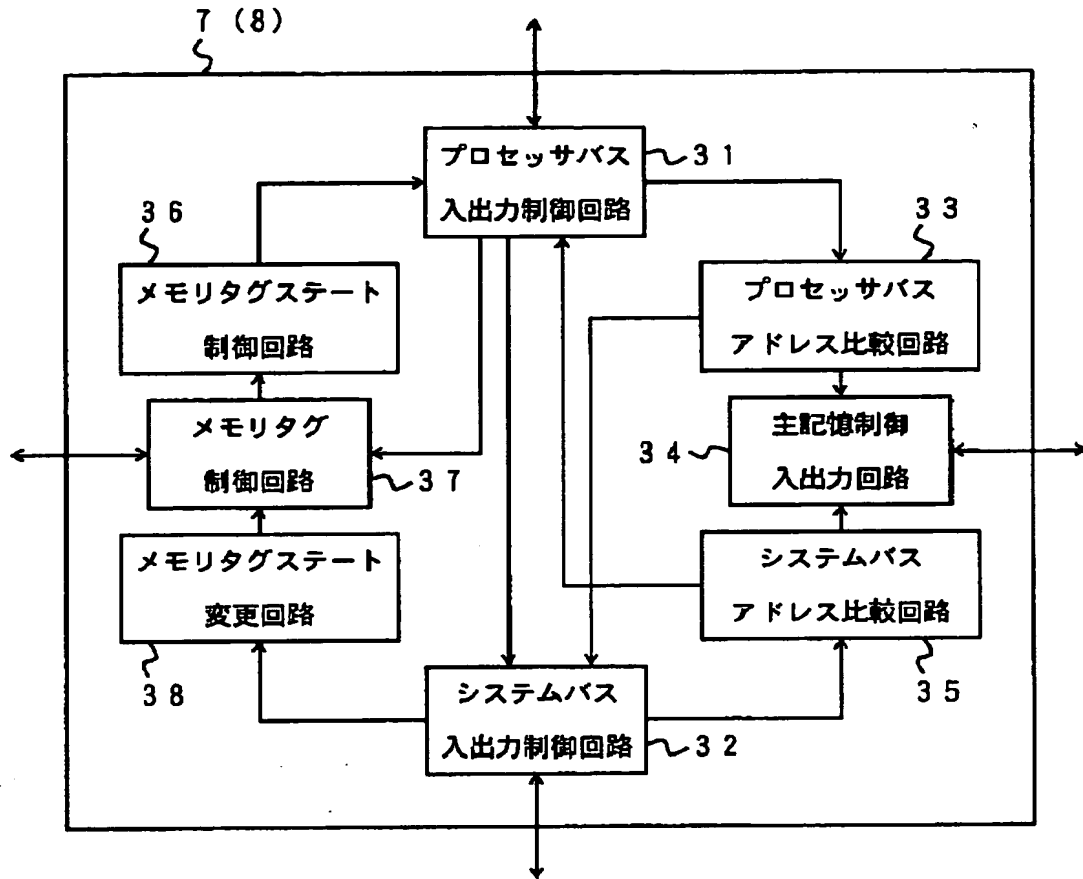
【図 1】



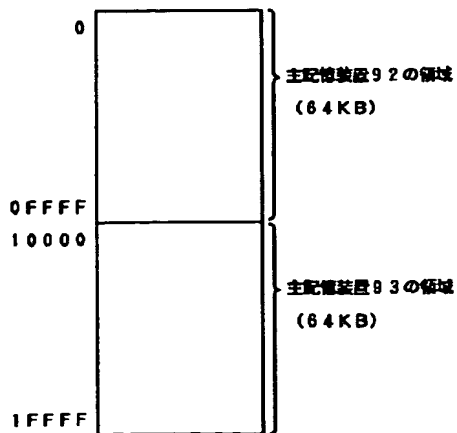
【図 4】



【図 3】



【図 5】



100 (主記憶アドレス)

0000
0040
0080
00C0
0100
≈
FF00
FF40
FF80
FFC0

→
→
→
→
→
≈
→
→
→
→

102 (メモリタグアドレス)

000
001
002
003
004
≈
3FC
3FD
3FE
3FF

タグの意味
 "0" : メモリデータが無効 (Invalid)
 (Modified data が何れかのキャッシュに有り)
 "1" : メモリデータが有効 (Valid)
 (Modified data が何れかのキャッシュにも無し)

```

graph TD
    Invalid((Invalid)) -- "load miss [RB]" --> Shared((Shared))
    Invalid -- "load hit, RBのスヌープ" --> Invalid
    Invalid -- "Store miss [RWITM]" --> Modified((Modified))
    Invalid -- "置換え, DC, RWITMのスヌープ" --> Shared
    Shared -- "置換え, RBのスヌープ [WB]" --> Invalid
    Shared -- "load hit, RBのスヌープ" --> Shared
    Shared -- "Store hit [DC]" --> Modified
    Modified -- "置換え, RWITMのスヌープ [WB]" --> Invalid
    Modified -- "置換え, RBのスヌープ [WB]" --> Shared
    Modified -- "load hit, Store hit" --> Modified
  
```

RB: リード要求
RWITM: 無効化つきリード要求
DC: 無効化要求
WB: ライト要求

```
graph LR; Invalid((Invalid)) -- "-WBのスヌープ" --> Valid((Valid)); Valid -- "-DC, RWITMのスヌープ" --> Invalid; Invalid -- "Self-loop" --> Invalid; Valid -- "Self-loop" --> Valid;
```

Invalid

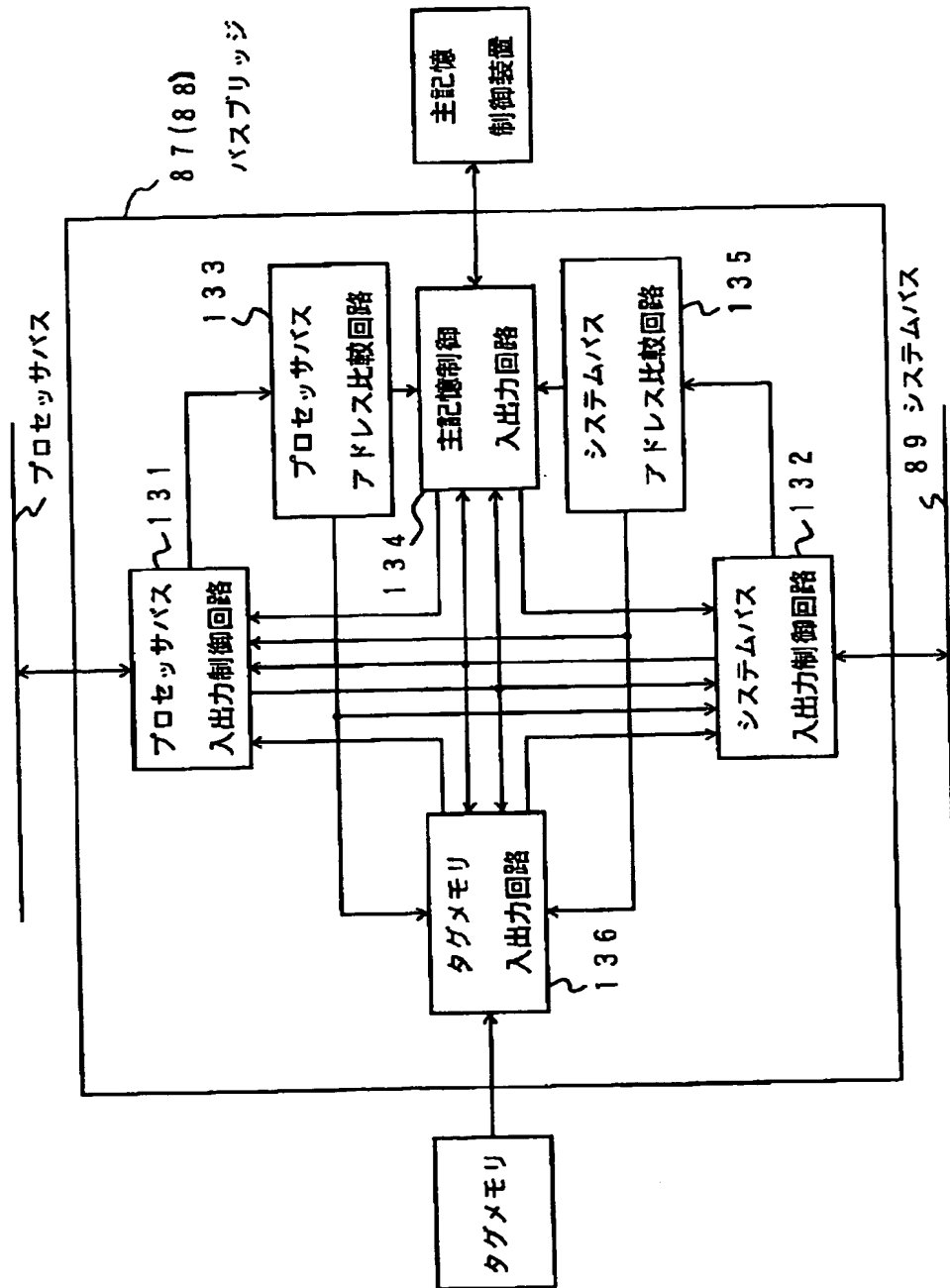
Valid

-WBのスヌープ

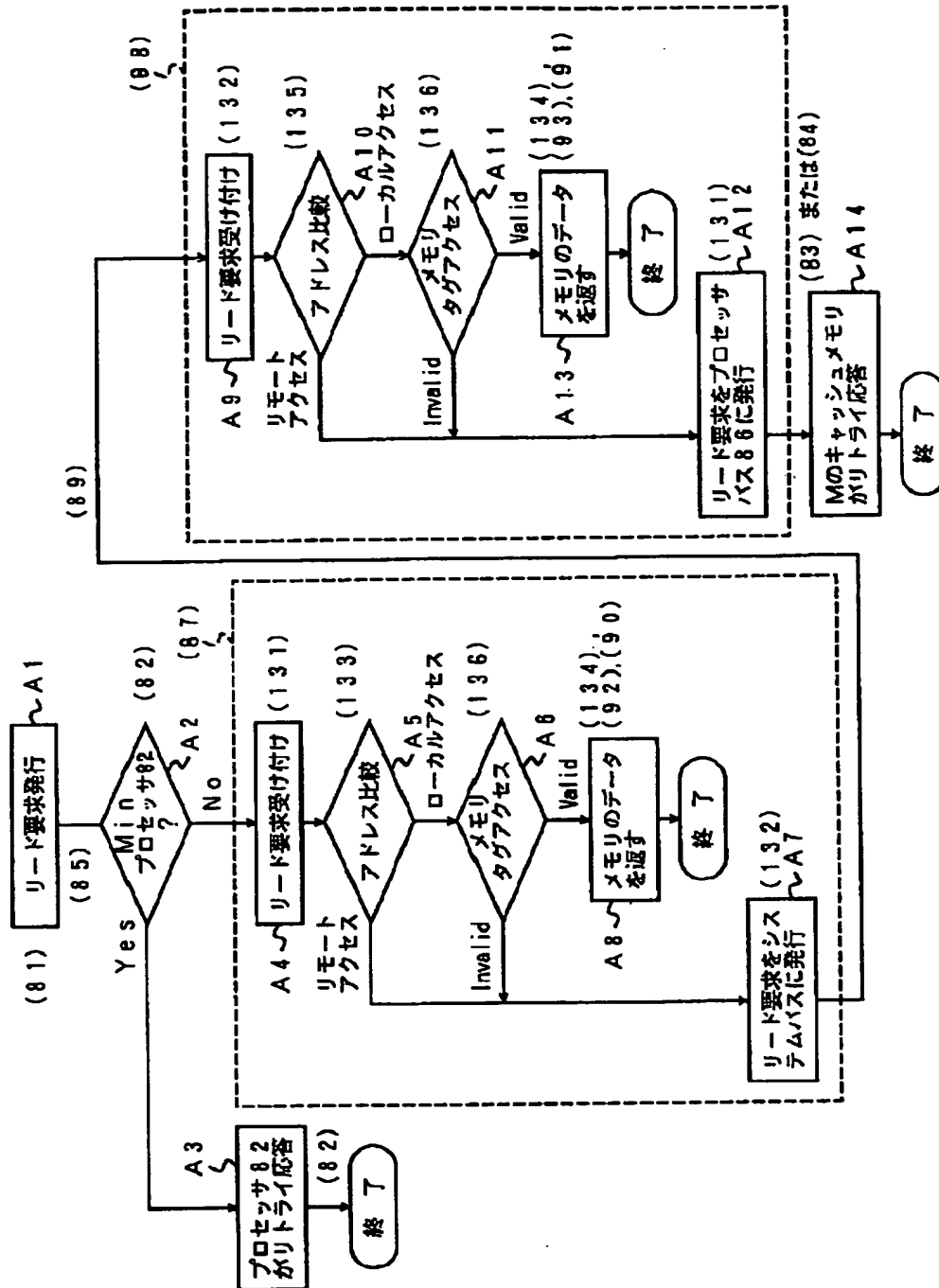
-DC, RWITMのスヌープ

RWITM: 無効化つきリード要求
DC: 無効化要求
WB: ライト要求

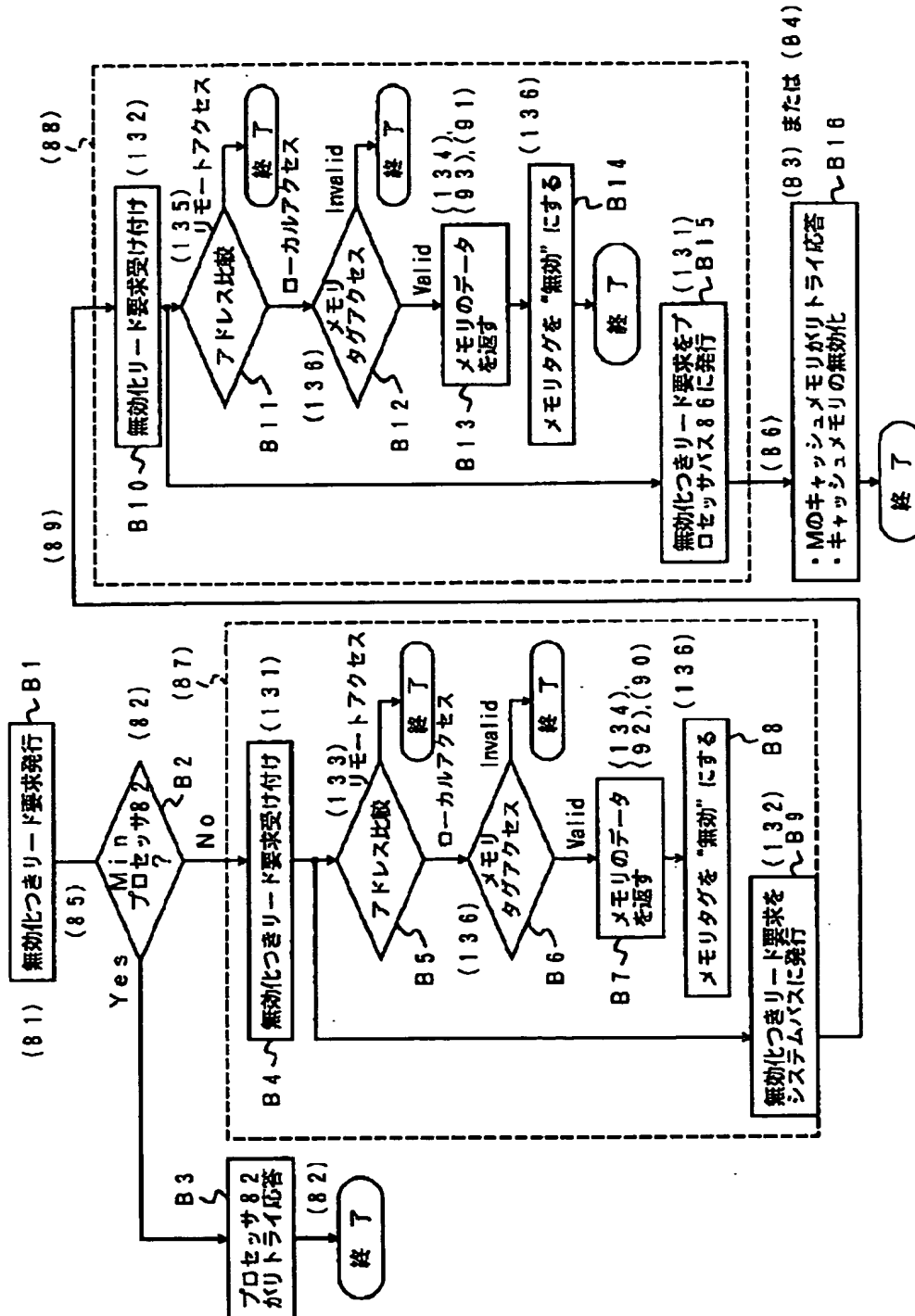
【図 7】



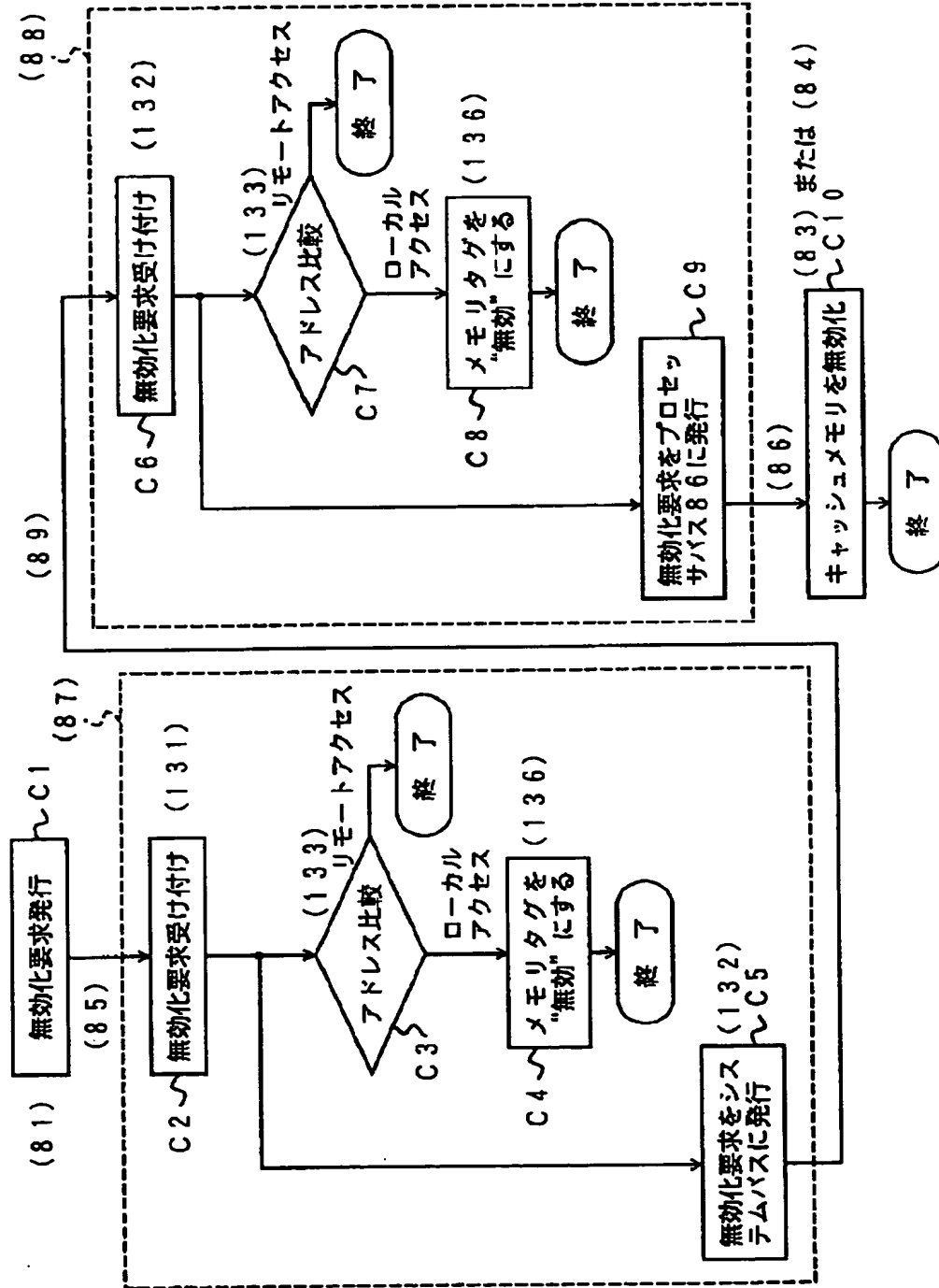
【図10】



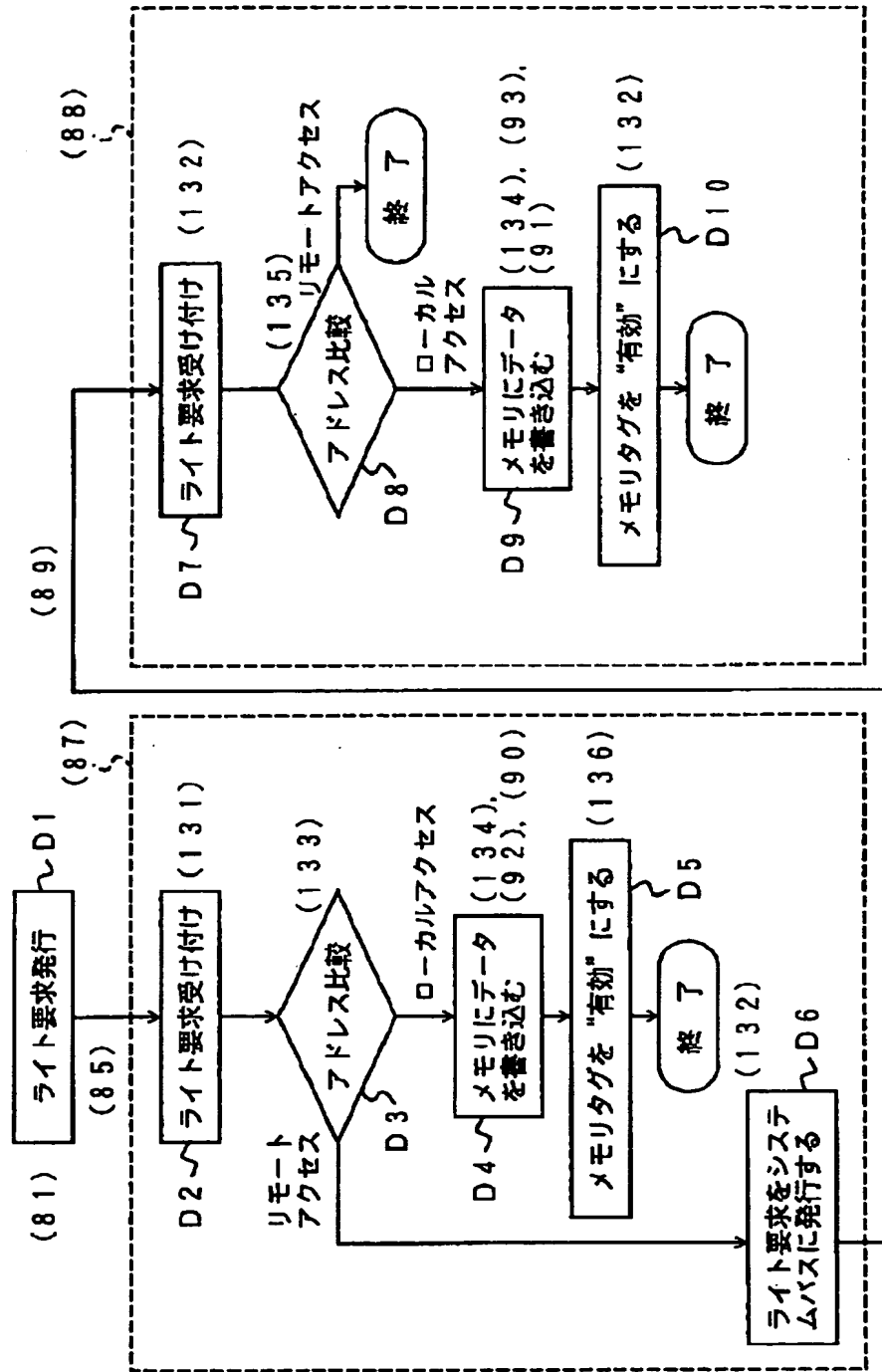
【図 11】



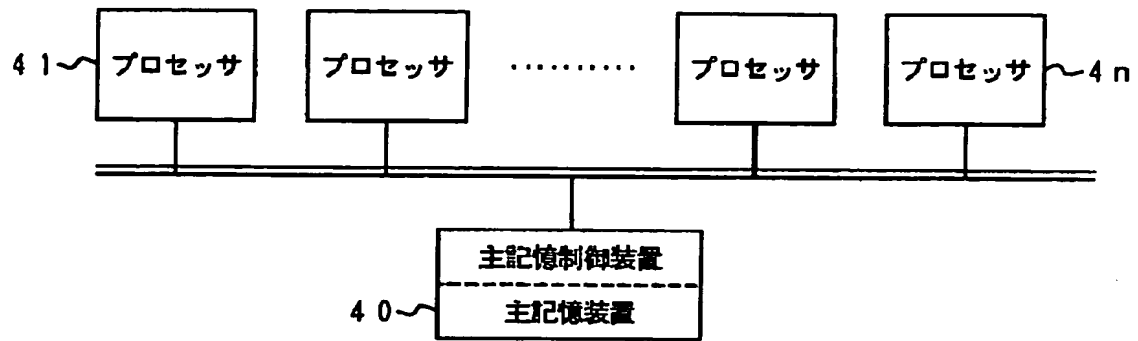
【図 12】



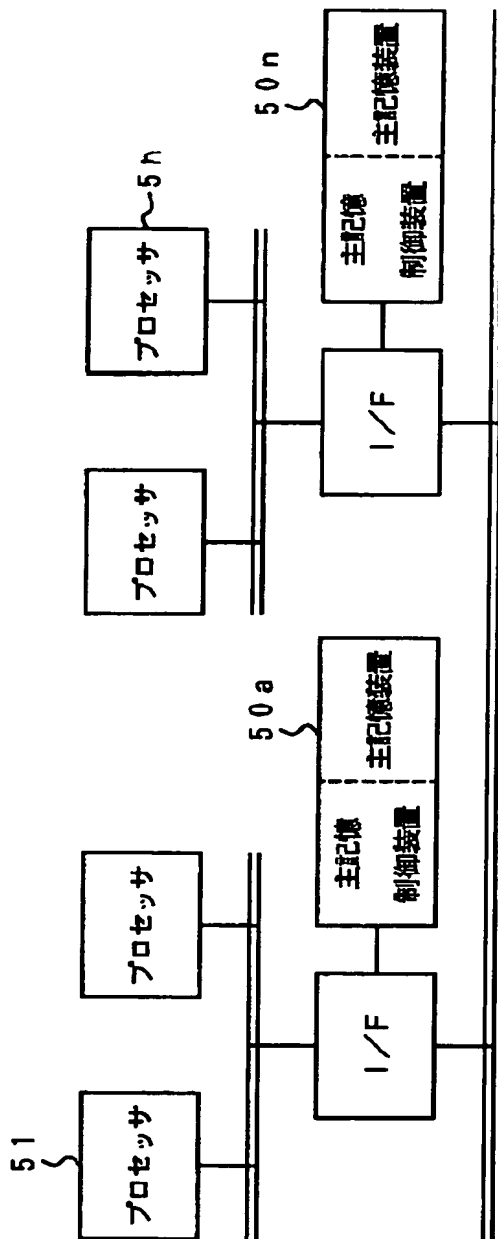
【図 13】



【図 1 4】



【図 15】



【図 16】

